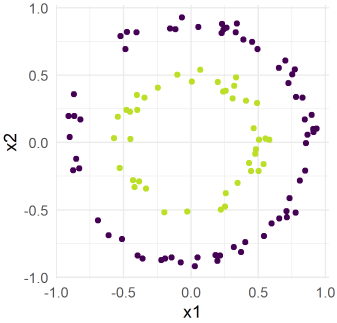
Introduction to Machine Learning

Details on Support Vector Machines

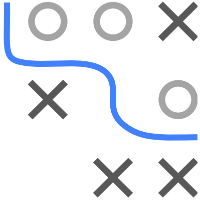


Learning goals

- Know that SVMs are non-parametric models
- Understand the concept of universal consistency
- Know that SVMs with an universal kernel (e.g. Gaussian kernel) are universally consistent

SVMs as Non-Parametric Models

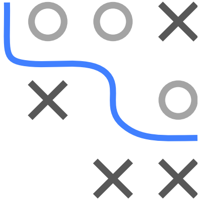
SVMS AS NON-PARAMETRIC MODELS



- In contrast to linear models, for an SVM we do not have to decide the number of coefficients of the decision function before training.
- The number of coefficients depends on the size of the dataset, or on the number of support vectors.
- Such models are called **non-parametric**.
- The big advantage of non-parametric models is that their modeling capacity is not *a priori* restricted to a finite-dimensional subspace of a function space.
- It turns out that SVMs do even better: There exist kernels so that an SVM can model all continuous functions arbitrarily well. It is also known that the SVM learning algorithm can approximate the Bayes optimal decision function arbitrarily well in the limit of infinite data.
- This property is known as **universal consistency**.

SVMS AS NON-PARAMETRIC MODELS

Definition [Steinwart, 2002]: Let $\mathcal{X} \subset \mathbb{R}^p$ be compact. A continuous kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called universal if the set of all induced functions $\sum_i \beta_i k\left(\mathbf{x}^{(i)}, \cdot\right)$ is dense in $\mathcal{C}(\mathcal{X})$; i.e., for all $g \in \mathcal{C}(\mathcal{X})$ and all $\varepsilon > 0$ there exists a function f induced by k with $\|f - g\|_\infty \leq \varepsilon$.



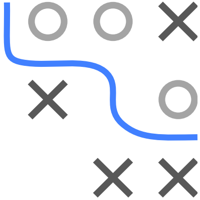
Example: Gaussian kernels are universal.

Theorem [simplified from Steinwart, 2002]: For compact $\mathcal{X} \subset \mathbb{R}^p$ define $C(n) = C_0 \cdot n^{q-1}$ for some $C_0 > 0$ and $0 < q < 1/p$. Fix any distribution $\mathbb{P}$ on $\mathcal{X} \times \{\pm 1\}$ from which i.i.d. datasets $\mathcal{D}_n$ of size n are drawn. Let h_n denote the soft-margin SVM model, trained with a universal kernel and regularization constant $C(n)$ on the data $\mathcal{D}_n$. Then it holds

$$\lim_{n \to \infty} \mathbb{E}[\mathcal{R}(h_n)] = \mathcal{R}^* \quad ,$$

where $\mathcal{R}^*$ denotes the Bayes risk.

ASYMPTOTIC PERFORMANCE



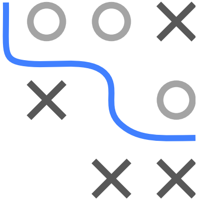
- Convergence of the risk to the Bayes risk for all distributions is called **universal consistency**.
- A universally consistent learning machine can solve all problems optimally, provided enough data.
- Parametric models are too inflexible for this property. They can model only a finite-dimensional subspace (manifold) of decision functions.
- Thus, in the limit of infinite data, they will systematically underfit.
- Universal consistency requires more than infinite-dimensional modeling power: We also need a learning rule that uses the flexibility wisely and avoids overfitting.
- The existence of universally consistent learners is one of the most exciting facts from non-parametric statistics.

ASYMPTOTIC PERFORMANCE

- Note the arbitrary positive constant C_0 in the definition of $C(n) = C_0 \cdot n^{q-1}$.
- This means that for a single fixed n, $C(n)$ can have any positive value.
- This is not a problem for the theorem since all it requires is that C changes at the right rate with n:
  - $n \cdot C(n)$ tends to infinity, which means that the relative impact of the regularizer compared to the empirical risk decays to zero, so, the risk term takes over for large n;
  - The convergence of $n \cdot C(n)$ to infinity is slow enough to avoid overfitting (this is far from obvious, but it is in the details of the proof of the theorem).
- Importantly, since C can be arbitrary for fixed n, this theorem does not tell us which C to use for a given problem size.

Kernels on Infinite-Dimensional Vector Spaces

KERNELS ON INFINITE-DIMENSIONAL VECTOR SPACES



- Note that the input space $\mathcal{X}$ does not need to be a finite-dimensional vector space.
- $\mathcal{X}$ could be the set of all character strings (of unlimited length) or of graphs, or of trees.
- Such data structures are natural representations for, e.g, HTML documents.
- There are many examples of data that do not naturally come in vector form.
- Most often meaningful and cheap-to-compute kernels can be defined directly on the input data structures – they simply define a similarity measure over these data.
- SVMs (and other kernel methods) allow to learn and predict directly on these spaces.